

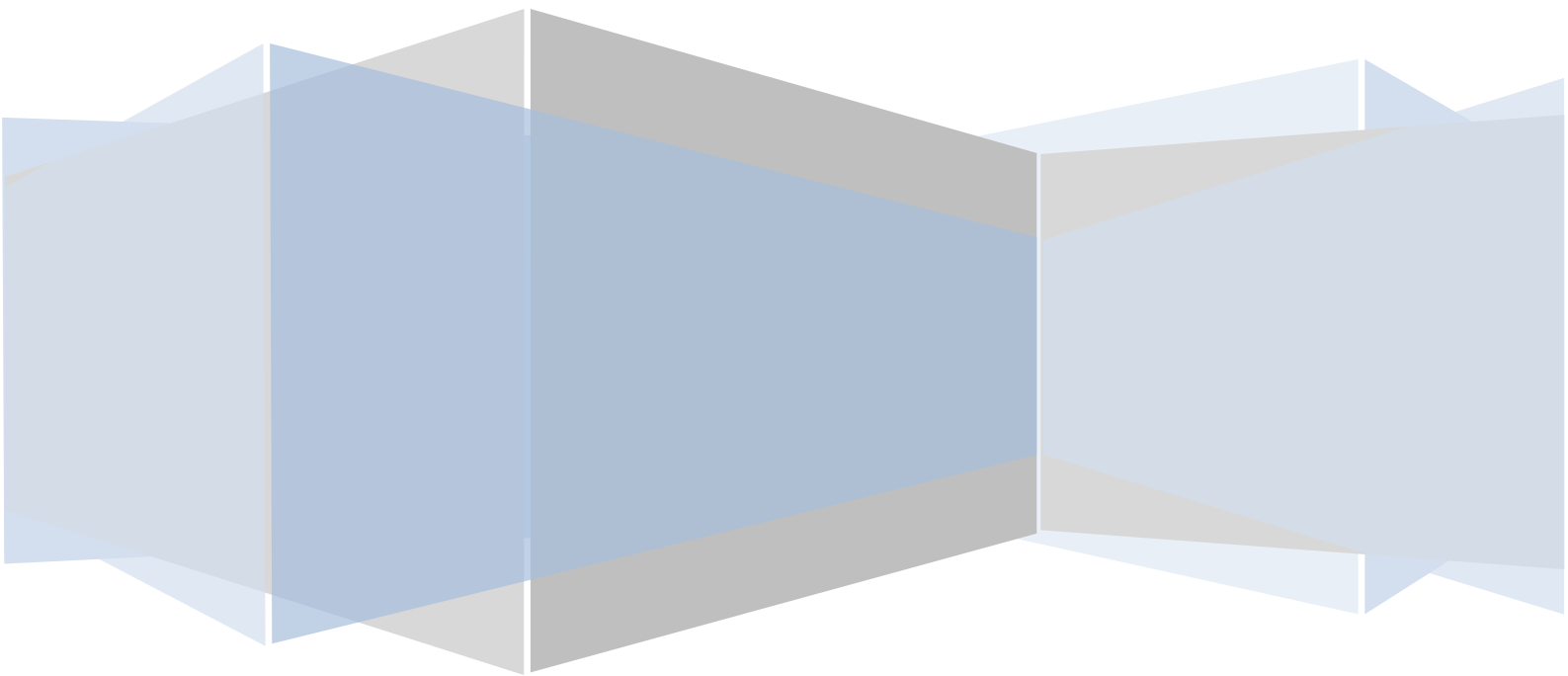
SSO Plugin

HP Service Request Catalog

J System Solutions

<http://www.javasystemsolutions.com>

Version 3.5



JSS SSO Plugin - Service Request Catalog

Introduction.....	3
Known issues.....	3
Direct URLs.....	3
Manual login.....	3
Enabling the identity federation service.....	4
Federation key.....	4
Token lifetime.....	4
Enabling Service Request Catalog.....	5
Copying files to SRC.....	5
Updating the web.xml file.....	5
Updating the applicationContext.properties file.....	6
Enabling the manual login page.....	6
Testing the integration.....	7

Introduction

The SSO Plugin provides a range of comprehensive range of SSO capabilities for HP Service Manager. This document explains how to deploy SSO Plugin to HP Service Request Catalog using HP Service Manager running SSO Plugin.

Please ensure the same version of SSO Plugin is deployed to both components.

Known issues

There are a couple of known issues when implementing SSO for SRC and these have been raised with HP's SRC product development team.

Direct URLs

The approach outlined in this document is suitable for users who do not wish to directly link to SRC entries. The direct link URL looks like this:

<http://server2k3.localdomain.local:8080/src/secure/main.jsp#account/serviceRequest/SD10317>

and the identifier (ie the part after the #) is being used to carry parameters. This information does not survive the SSO process outlined in this document and HP have accepted that the URL should not use an identifier, ie should have a format similar to this:

<http://server2k3.localdomain.local:8080/src/account/serviceRequest/SD10317>

We have been informed this change will be made in an update to version 1.3, or in 1.4 (due late 2012). If you require direct linking functionality, please consult the second SRC installation guide for an alternative approach.

Manual login

The manual login page does not work when SSO is enabled, no matter what SSO product is installed. HP have confirmed this to be an issue and have informed us it will be fixed in SRC version 1.4.

Enabling the identity federation service

The identity federation service isn't enabled by default. To enable it, go to the SSO Plugin setup page and click "Enable identity federation service". The following fields must also be set.

Federation key

This is known only to the SSO Plugin and the third party applications that are integrating with the SSO Plugin. It is used as a seed to a hashing function that allows the identity federation service and acceptor to trust each other.

Token lifetime

When a user has been authenticated, the JSS identity federation acceptor sets a cookie (with the name `jss-ssoplugin`). The cookie contains an encoded token that includes the logged in user. The cookie is set to expire after a period of time that is calculated to be the time at which the user was authenticated with SSO Plugin, plus the value defined in this field.

Typically, a value of one day will ensure every client has to re-authenticate with SSO Plugin once a day.

Enabling Service Request Catalog

There are three steps to this process: Copying a jar file to SRC, updating the SRC web.xml file so it will use SSO Plugin for authentication and updating the SRC applicationContext.properties file.

Prior to performing the steps, stop the Java web server (ie Tomcat) instance running Service Request Catalog. When the steps have been completed, start it again.

Copying files to SRC

Locate the jss-ss0-thirdparty.jar file in the src directory within the SSO Plugin installation set. Copy it to the SRC WEB-INF/lib directory, found in the SRC program directory (ie tomcat/webapps/src/WEB-INF/lib).

Updating the web.xml file

Locate the SRC web.xml file, found in the SRC program directory (ie tomcat/webapps/src/WEB-INF/web.xml). Make a backup of this file.

Open the file and locate the following:

```
<context-param>
  <param-name>contextClass</param-name>
  <param-value>
    com.hp.service.catalog.server.web.context.CustomXmlWebApplicationContext
  </param-value>
</context-param>
```

Locate the web.xml.src.patch file within the installation set src directory. Open it in a text editor, copy it all and paste into the web.xml file placing below the </context-param> specified above in bold.

In the patch you have just applied, locate the following two parameters and set appropriately:

1. **identityFederationServiceURL:** This points to the identity federation service running on the SSO Plugin installation. The identity federation service URL is /jss-ss0/identityfederationservice, relative to the SSO Plugin enabled HP Service Manager Web Tier. Therefore, if SSO Plugin is installed on the Web Tier at this URL:

```
http://servicemanager:8080/webtier
```

then the identityFederationServiceURL is:

```
http://servicemanager:8080/webtier/jss-ss0/identityfederationservice
```

2. **key:** This must be set to the federated identity key set in the SSO Plugin interface.

The **logLevel** parameter is optional and controls the amount of logging generated by the identity federation acceptor, the valid values are INFO, DEBUG and TRACE. You are advised to set this to INFO in production.

Updating the applicationContext.properties file

Locate the SRC applicationContext.properties file, found in the SRC program directory (ie tomcat/webapps/src/WEB-INF/classes/applicationContext.properties).

This file is the SRC configuration file that is modified to specify the location of Service Manager. It also specifies the authentication method. Locate the following:

```
# Security Mode: Choose your security method  
and select:
```

```
src.security.mode=remoteUsrSsoUiAndTsoWs
```

Enabling the manual login page

Unfortunately, the SRC application combines a login page with the main application, so it is difficult to separate SSO and non-SSO access given the main application is protected via SSO.

To work around this problem, the following steps will result in the logout process presenting a login page:

1. Rename the logout.jsp to logout.jsp.old.
2. Using your favourite text editor, create a new logout.jsp with the following content:

```
<% response.sendRedirect(request.getContextPath()  
+ "/secure/nosso.jsp"); %>
```

3. Copy the src/secure/main.jsp to src/secure/nosso.jsp.
4. Locate the SRC applicationContext-security-remoteUsrSsoUiAndTsoWs.xml file, found in the SRC program directory (ie tomcat/webapps/src/WEB-INF/spring/security/applicationContext-security-remoteUsrSsoUiAndTsoWs.xml). Open the file in a text editor, find the section below and add the text in bold:

```
<http entry-point-ref="authenticationProcessingFilterEntryPoint">  
  <intercept-url pattern="/logout.jsp" filters="none" />  
  <intercept-url pattern="/secure/nosso.jsp" filters="none" />
```

Testing the integration

When installation is complete, ensure SSO is functioning by visiting the SSO Plugin Test SSO page running on Service Manager, ie. <http://smserver/webtier/jss-ss0/testss0.jsp>.

Assuming SSO functions, and most importantly, your SSO user has a valid SSO account in Service Manager, navigate to SRC. The browser should sign in to SRC as the SSO user reported by the Test SSO page.